D. Remarks

Based on the above amendments and remarks to follow, reconsideration of this application is respectfully requested.

In the office action, claims 1-12 and 24-26 were rejected under 35 U.S.C. 102(e) as being anticipated by Gilboa (US2004/0148586A1). Claims 13-23 were withdrawn from further consideration pursuant to 37 CFR 1.142(b), as being drawn to a nonelected invention. Further, in claim 1, the phrase "flowchart like manner" was said to render the claim indefinite under 35 USC 112.

Furthermore, the specification was objected to as failing to provide proper antecedent basis for the claimed subject matter. Claims 24-26 were rejected on the ground that the specification did not provide a basis for the "computer program product recitation". In order to correct this, a paragraph has been added to the specification which is based on the recitations of claims 24-26 *as filed*. As the original claims form part of the specification, this added paragraph does not constitute new matter.

In response to the objections and rejections cited above, the specification has been amended. Further, claims 2 and 6-10 have been cancelled without any prejudice or disclaimer. Claims 1, 3-5, 11-12 and 24-26 have been amended and claims 27-28 have been newly added to remove any indefinteness and to more particularly define the invention.

Descriptions and Differences of the Claims from the Cited Art:

In order to more clearly define and distinctly claim the present invention from the cited art, and to particularly point out and distinctly claim the subject matter, independent claim 1 has been amended. The claim, as amended, recites a method for enabling an application designer and a user to develop a User Interface (UI) from UI models without coding (*claim 1, lines 1-3*). Support for these recitations is found in the present application on page 6, lines 16-19; page 20, lines 16-18; page 32, lines 23-27; and page 33, lines 1-9. *In contrast, Gilboa discloses the generation of the UI from UI models by translating the UI models to an executable code. In Gilboa, the code is generated automatically and not manually. The generation of the code is an important*

feature of Gilboa, whereas, in the present application, coding does not take place to develop a UI (claim 1, lines 1-3). Therefore, the phrase "creating UIs without manual coding" in Gilboa cannot be construed as "developing a UI without coding".

As set forth in independent claim 1, a User Interface (UI) is developed from UI models without coding. The UI models are developed by using pre-built reusable components *(claim 1, lines 1-4)*. Support for these recitations is found in the present application on page 6, lines 7-9; page 6, lines 23-27; and page 7, lines 1-3. The application designer identifies processes according to the requirements of the UI *(claim 1, clause (a), lines 1-2)*. Support for this recitation is found in the present application on page 10, lines 18-22 and page 12, lines 23-25. The tasks required to define the identified processes are defined. These tasks are defined by providing the meta-data to instances of a set of pre-built reusable components by using a visual modeling environment. This pre-built reusable component is an abstract object that is built to perform a function *(claim 1, clause (b), lines 1-5)*. Support for these recitations is found in the present application on page 7, lines 2-3 and page 12, lines 11-22. Further, the defined tasks are connected in a logical order by using the visual modeling environment. The defined tasks are connected to model the identified processes, which are used to develop UI models *(claim 1, clause (c), lines 1-4)*. Support for these recitations is found in the present application on page 7, lines 5-7; page 12, lines 30-31; page 13, lines 1-3; page 20, lines 1-2 and page 20, lines 10-15. The developed UI models are stored in a database *(claim 1, clause (d), lines 1-2)*. Support for this recitation is found in the present application on page 7, line 1 and page 18, lines 23-24.

In the present invention, the UI models are executed by an engine. The execution is based on requests that are made by the application designer and the user while developing the UI. These requests are concurrently executed by loading the identified processes from the database and executing the tasks specified in the identified processes in the logical order *(claim 1, clause (e), lines 1-6)*. Support for these recitations is found in the present application on page 12, lines 6-9; page 18, lines 10-15 and page 33, lines 2-13.

On the other hand, Gilboa discloses only a modeling system or a visual tool for creating UI models. Gilboa also discloses gathering the user requirements for a particular context of the UI. Further, Gilboa discloses defining user tasks according to the user requirements by assigning specific properties to scenes and scenarios. Furthermore, Gilboa discloses creating UI models by defining the relationship between the different scenes and storing UI models in a repository. Gilboa only discloses generation of the UI from UI models by translating UI models to an executable code. *However, Gilboa does not disclose developing a UI by executing the plurality of UI models, as set forth in claim 1, clause (e), lines 1-3 of the present application. Further, Gilboa does not disclose an engine that executes the UI models and concurrently executes the requests made by the application designer and the user while developing the UI, as set forth in claim 1, clause (e), lines 1-4 of the present application.* Furthermore, the UI models in the present invention are executed by directly loading processes and executing the tasks specified in the processes in a logical order (*claim 1, clause (e), lines 1-6). In contrast, Gilboa discloses generation of the UI by generating an executable code from the canonical representation of the UI model.* Therefore, the method for developing a UI from UI models in the present invention is completely distinct from that disclosed in Gilboa.

Claim 3 has been amended in light of the amendments made to independent claim 1, and to clarify the correspondence between the language of the claim and that of the specification. Support for this recitation is found in the present application on page 20, line 30 and page 21, lines 1-13.

Claim 4 has been amended in light of the amendments made to independent claim 1, and to clarify the correspondence between the language of the claim and that of the specification. Support for this recitation is found in the present application on page 7, lines 2-7 and page 12, lines 14-22.

Claim 5 has been amended in light of the amendments made to independent claim 1, and to clarify the correspondence between the language of the claim and that of the specification. Support for this recitation is found in the present application on page 7, lines 14-17.

Claim 11 has been amended in light of the amendments made to independent claim 1. The claim, as amended, illustrates the method for executing UI models to develop the UI. The application designer and the user input requests while developing the UI. These requests are transferred to an engine for processing. The pre-built reusable components that are required to process these requests are identified and cached (*claim 11, lines 1-10)*. The tasks that are defined in the processes are executed in a logical order. The errors that occur during processing of requests are handled and the information related to the execution of the tasks is logged in the database. Furthermore, the results of the execution are output (*claim 11, lines 11-17)*. Support for these recitations is found in the present application on page 12, lines 6-9; page 32, lines 23-27; page 33, lines 1-9; page 34, lines 12-18; page 36, lines 1-11; page 36 lines 27-30 and page 37, lines 1-3.

In contrast, Gilboa discloses the execution of UI models to generate an executable code. *Gilboa does not disclose the execution of the UI models by processing user requests and by directly executing tasks in a logical order,* as set forth in *claim 11, lines 1-2; claim 11, lines 11-12* of the present application. Further, in the present application, the requests are made by the application designer while developing the UI models and by the user while executing the UI models. These requests are transferred to the engine for processing *(claim 11, lines 4-7)*. In contrast, Gilboa discloses gathering user requirements to develop UI models. *Furthermore, the engine of the present invention processes the requests made while the UI is being developed (claim 11, line 7). In contrast, Gilboa discloses a classification engine that organizes contents in folders or tree structures. Moreover, the present invention discloses the identification of the pre-built reusable components that are required to process the requests (claim 11, lines 8-9). In contrast, Gilboa discloses the identification of the information services and objects that are required to build the information model. Therefore, the identification of the pre-built reusable components in the present invention cannot be construed as the identification of information services and objects in Gilboa.*

Further, *Gilboa does not disclose caching of pre-built reusable components that are required to process requests, as set forth in claim 11, lines 8-10 of the present*

application. Gilboa only discloses storing of the UI models in a repository. Furthermore, *Gilboa does not disclose handling of errors that occur while processing requests,* as set forth in *claim 11, lines 13-14* of the present application. Moreover, Gilboa does not *disclose logging of information related to the execution of tasks, as* set forth in *claim 11, line 15* of the present application. Gilboa only discloses storing of a unification object model in a repository while UI models are being developed.

To more clearly define and distinctly claim the present invention from the cited art, and to particularly point out and distinctly claim the subject matter, independent claim 12 has been amended. The claim, as amended, recites a method for enabling an application designer and a user to develop a User Interface (UI) from UI models without coding *(claim 12, lines 1-3)*. Support for these recitations is found in the present application on page 6, lines 16-19; page 20, lines 16-18; page 32, lines 23-27 and page 33, lines 1-9. *In contrast, Gilboa discloses the generation of the UI from UI models by translating the UI models to an executable code. In Gilboa, the code is generated automatically and not manually. The generation of the code is an important feature in Gilboa, whereas, in the present application, coding does not take place to develop a UI (claim 12, lines 1-3). Therefore, the phrase "creating UIs without manual coding" in Gilboa cannot be construed as "developing a UI without coding".*

As set forth in independent claim 12, a User Interface (UI) is developed from UI models without coding. The UI models are developed by using pre-built reusable components *(claim 12, lines 1-4)*. Support for these recitations is found in the present application on page 6, lines 7-9; page 6, lines 23-27 and page 7, lines 1-3. The application designer identifies processes according to the requirements of the UI *(claim 12, clause (a), lines 1-2)*. Support for this recitation is found in the present application on page 10, lines 18-22 and page 12, lines 23-25. The tasks required to define the identified processes are defined. These tasks are defined by providing the meta-data to instances of a set of pre-built reusable components by using a visual modeling environment. This pre-built reusable component is an abstract object that is built to perform a function (*claim 12, clause (b), lines 1-5).* Support for these recitations is found in the present application on page 7, lines 2-3 and page 12, lines 11-22. Further, the defined tasks are verified by applying a set of pre-defined

verifications on each of the defined tasks *(claim 12, clause (c), lines 1-2)*. Support for this recitation is found in the present application on page 22, lines 3-8. Furthermore, the defined tasks are connected in a logical order by using the visual modeling environment. The defined tasks are connected to model the identified processes, which are used to develop UI models *(claim 12, clause (d), lines 1-4)*. Support for these recitations is found in the present application on page 7, lines 5-7; page 12, lines 30-31; page 13, lines 1-3; page 20, lines 1-2 and page 20, lines 10-15. The developed UI models are stored in a database (*claim 12, clause (e), line 1)*. Support for this recitation is found in the present application on page 7, line 1 and page 18, lines 23-24.

The application designer and the user input requests while developing the UI (*claim 12, clause (f), lines 1-2)*. These requests are transferred to an engine for processing (*claim 12, clause (g), line 1)*. The pre-built reusable components that are required to process the requests are identified and cached (*claim 12, clause (h), lines 1-2 and claim 12, clause (i), line 1)*. The tasks that are defined in the processes are executed in a logical order (*claim 12, clause (j), line 1)*. The errors that occur during processing of requests are handled and information related to the execution of tasks is logged in the database (*claim 12, clause (k), lines 1-2 and claim 12, clause (l), line 1)*. Further, the results of the execution are output (*claim 12, clause (m), line 1)*. Support for these recitations is found in the present application on page 12, lines 6-9; page 32, lines 23-27; page 33, lines 1-9; page 34, lines 12-18; page 36, lines 1-11; page 36 lines 27-30 and page 37, lines 1-3.

In contrast, Gilboa discloses the execution of UI models to generate an executable code. *Gilboa does not disclose the execution of UI models by processing user requests and by directly executing tasks in a logical order, as set forth in claim 12, lines 1-2 and claim 12, and clause (j), line 1* of the present application. Further, in the present application, the requests are made by the application designer while developing the UI models and by the user while executing the UI models. These requests are transferred to the engine for processing *(claim 12, clause (f), lines 1-2 and claim 12, clause (g), line 1)*. In contrast, Gilboa discloses gathering of user requirements to develop UI models. *Furthermore, the engine of the present invention*

processes the requests made while the the UI was being developed (claim 12, clause (g), line 1). In contrast, Gilboa discloses a classification engine that organizes contents in folders or tree structures. Moreover, the present invention as defined in claim 12 recites the identification of the pre-built reusable components that are required to process the requests (claim 12, clause (h), lines 1-2). In contrast, Gilboa discloses the identification of the information services and objects that are required to build the information model. Therefore, identification of the pre-built reusable components in the present invention cannot be construed as an identification of the information services and objects in Gilboa.

Gilboa does not disclose caching of pre-built reusable components that are required to process requests, as set forth in claim 12, clause (i), line 1 of the present application. Gilboa only discloses storing of UI models in a repository. Further, *Gilboa does not disclose handling of errors that occur while processing requests,* as set forth in *claim 12, clause (k), lines 1-2* of the present application. *Moreover, Gilboa does not disclose logging of information related to the execution of tasks, as set forth in claim 12, clause (l), line 1 of the present application.* Gilboa only discloses storing of a unification object model in a repository while UI models are being developed.

To more clearly define and distinctly claim the present invention from the cited art, and to particularly point out and distinctly claim the subject matter, independent claim 24 has been amended. The claim, as amended, recites a computer program product that is for use with a computer. The computer program product includes a usable medium with a computer readable program code that is embodied in it. The computer readable program code enables an application designer and an application user to develop a User Interface (UI) from UI models without coding (claim 24, lines 3-6). Support for these recitations is found in the present application on page 6, lines 16-19; page 20, lines 16-18; page 32, lines 23-27 and page 33, lines 1-9. *In contrast, Gilboa discloses the generation of the UI from UI models by translating the UI models to an executable code. In Gilboa, the code is generated automatically and not manually. The generation of the code is an important feature of Gilboa, whereas in the present application, coding does not take place to develop a UI (claim 24, lines 3-6).*

Therefore, the phrase "creating UIs without manual coding" in Gilboa cannot be construed as "developing a UI without coding".

As set forth in independent claim 24, a User Interface (UI) is developed from UI models without coding. The UI models are developed by using pre-built reusable components *(claim 24, lines 4-7)*. Support for these recitations is found in the present application on page 6, lines 7-9; page 6, lines 23-27 and page 7, lines 1-3. The application designer identifies processes according to the requirements of the UI *(claim 24, clause (a), lines 1-2)*. Support for this recitation is found in the present application on page 10, lines 18-22 and page 12, lines 23-25. The tasks that are required to define the identified processes are defined by providing the meta-data to instances of a set of pre-built reusable components by using a visual modeling environment. This pre-built reusable component is an abstract object that is built to perform a function (*claim 24, clause (b), lines 1-5)*. Support for these recitations is found in the present application on page 7, lines 2-3 and page 12, lines 11-22. The defined tasks are connected in a logical order by using the visual modeling environment. The defined tasks are connected to model the identified processes, which are used to develop UI models *(claim 24, clause (c), lines 1-4)*. Support for these recitations is found in the present application on page 7, lines 5-7; page 12, lines 30-31; page 13, lines 1-3; page 20, lines 1-2 and page 20, lines 10-15. The developed UI models are stored in a database (*claim 24, clause (d), lines 1-2)*. Support for this recitation is found in the present application on page 7, line 1 and page 18, lines 23-24.

In the present invention, the UI models are executed by an engine. This execution is based on requests made by the application designer and the user while developing the UI. These requests are concurrently executed by loading the identified processes from the database and executing the tasks specified in the identified processes in the logical order (*claim 24, clause (e), lines 1-6)*. Support for these recitations is found in the present application on page 12, lines 6-9; page18, lines 10-15 and page 33, lines 2-13.

On the other hand, Gilboa discloses only a modeling system or a visual tool for creating UI models. Gilboa discloses gathering the user requirements for a particular context of the UI. Further, Gilboa discloses defining user tasks according to the user requirements by assigning specific properties to scenes and scenarios. Furthermore, Gilboa discloses the creation of UI models by defining the relationship between different scenes and storing UI models in a repository. Gilboa only discloses the generation of the UI from UI models by translating the UI models to an executable code. *However, Gilboa does not disclose a UI being developed by executing the plurality of UI models,* as set forth in *claim 24, clause (e), lines 1-3* of the present application. Moreover, *Gilboa does not disclose an engine that executes UI models and concurrently executes the requests made by the application designer and the user while developing the UI, as set forth in claim 24, clause (e), lines 1-4 of the present application.* Further, the UI models in the present invention are executed by directly loading processes and executing the tasks specified in the processes in a logical order (*claim 24, clause (e), lines 1-6). In contrast, Gilboa discloses the generation of the UI by generating an executable code from the canonical representation of the UI model.* Therefore, the method for developing the UI from UI models in the present invention is different from that disclosed in Gilboa.

Claim 25 has been amended in light of the amendments made to independent claim 24. Support for these recitations is found in the present application on page 22, lines 3-8.

Claim 26 has been amended in light of the amendments made to independent claim 24. The claim, as amended, recites a computer program product that includes a program code, which is embodied in it, to enable users to execute the UI models. The application designer and the user input requests while developing the UI. These requests are transferred to an engine for processing. The pre-built reusable components that are required to process these requests are identified and cached. The tasks that are defined in the processes are executed in a logical order. Errors that occur during the processing of requests are handled and information related to the execution of the tasks is logged in the database. Furthermore, the results of the execution are output (*claim 26, lines 1-19).* Support for these recitations is found in the

present application on page 12, lines 6-9; page 32, lines 23-27; page 33, lines 1-9; page 34, lines 12-18; page 36, lines 1-11; page 36 lines 27-30 and page 37, lines 1-3.

In contrast, Gilboa discloses the execution of UI models to generate an executable code. *Gilboa does not disclose the execution of UI models by processing user requests by directly executing tasks in a logical order, as set forth in claim 26, lines 2-3 and claim 26, lines 13-14 of the present application.* Further, in the present application, the requests are made by the application designer while developing the UI models and by the user while executing the UI models. These requests are transferred to the engine for processing *(claim 26, lines 6-9)*. In contrast, Gilboa discloses gathering user requirements to develop UI models. *Furthermore, the engine of the present invention processes the requests made while the UI (claim 26, line 9) is being developed. In contrast, Gilboa discloses a classification engine that organizes the contents in folders or tree structures. Moreover, the present invention discloses the identification of the pre-built reusable components required to process the requests (claim 26, lines 10-11). In contrast, Gilboa discloses the identification of the information services and objects required to build the information model. Therefore, the identification of the pre-built reusable components in the present invention cannot be construed as an identification of the information services and objects in Gilboa.*

Gilboa does not disclose *caching of pre-built reusable components that are required to process requests, as set forth in claim 26, lines 10-12 of the present application.* Gilboa only discloses storing of the UI models in a repository. Further, *Gilboa does not disclose handling of errors that occur while requests are being processed, as set forth in claim 26, lines 15-16 of the present application.* Moreover, *Gilboa does not disclose logging of information related to the execution of tasks, as set forth in claim 26, lines 17-18 of the present application*. Gilboa only discloses storing of a unification object model in a repository while UI models are being developed.

Further, claim 27 is a new addition that is dependent on amended independent claim 1, to recite a method that enables the application designer to verify the defined tasks. The defined tasks are verified by applying a set of pre-defined verifications on

each of the defined tasks (*claim 27, lines 2-3*). Support for these recitations is found in the present application on page 22, lines 3-14.

Furthermore, claim 28 is a new addition that is dependent on newly added claim 27, to recite a method that enables the application designer to visually verify the developed UI (claim 28, line 2). The verification includes observing the values of the watch variables while executing the UI models. The watch variables are identified by the application designer (claim 28, clause (a), lines 1-3). The verification further includes stopping at each break point while UI models are being executed. The break points are set by the application designer (claim 28, clause (b), lines 1-2). At each of the break points, information related to the UI models is analyzed (claim 28, clause (c), lines 1-2). Support for these recitations is found in the present application on page 22, lines 15-22.

In contrast, *Gilboa does not disclose visual verification of the UI by identifying watch variables and setting break points*, as set forth in *claim 28, clause (a), lines 1-3 and claim 28, clause (b), lines 1-2* of the present application. Gilboa only discloses verifying UI models by generating the UI and comparing the UI with the users' requirements. However, *the present invention enables the application designer to indentify and observe the values of the watch variables (claim 28, clause (a), lines 1-3).* Further, the present invention enables the application designer to set break points. The application designer in the present invention stops at the break points and analyzes information related to the UI models *(claim 28, clause (b), lines 1-2 and claim 28, clause (c), lines 1-2).*

Conclusion

In light of the amendments made to independent claim 1, dependent claims 3-5, 11 and 27-28 are thus allowable. Further, in light of the amendments made to independent claim 24, dependent claims 25 and 26 are thus allowable. It is respectfully submitted that the amendments made to independent claims 1, 12 and 24 are sufficient to remove the rejections under 35 U.S.C. 102(e).

Therefore, in light of the above, and in view of the amendments made to independent claims 1, 12 and 24, dependent claims 3-5, 11 and 25-26, and the

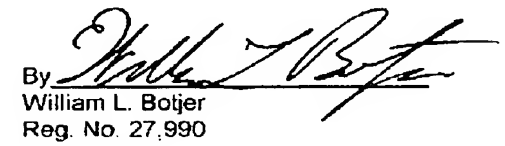
addition of the new claims 27-28, the present invention, as described in the present claims, is clearly patentable over Gilboa.

Further, amendments made to claim 1 are sufficient to remove the rejections under 35 U.S.C. 112. Furthermore, amendments made to the specification are sufficient to remove the objection under 37 CFR 1.75(d)(1) and MPEP §608.01(o).

The present claims have been amended to highlight the distinctions of the present invention over the cited art, and it is respectfully submitted that the claims are now clearly patentable over the art of record, and notice to that effect is earnestly solicited. If the examiner has any questions regarding this matter, the examiner is quested to telephone the applicant's attorney at the numbers listed below, prior to issuing a further action.

Dated: January 7, 2008

Respectfully Submitted,

By 

William L. Botjer
Reg. No. 27,990
PO Box 478
Center Moriches, NY 11934
(212) 737-5728 (Tue-Thurs)
(631) 874-4826 (Mon & Fri)
(631) 834-0611 (cell if others busy)